第5章 LSI検証プロセスをCMMに基づいて改善

── プロジェクト管理者の負担は増えたが，検証工程全体の見通しがよくなった

高橋 太

ここではソフトウェア開発のプロセス管理手法の一つであるCMM注の概要と，これをLSI検証プロセスに適用するときの考えかたを紹介する．筆者ら（富士通LSIテクノロジ）は全社的にCMMに取り組み，複数のプロジェクトでCMMに従ったプロセス改善活動を実施している．CMMを導入すると，プロジェクト・マネージャの負担は増加する．その一方で，検証工程全体の見通しはよくなった．例えば，開発プロセスの中のあいまいさが減ったり，プロジェクト間の品質のばらつきが小さくなったという．（編集部）

シミュレーションを利用する大規模ディジタルLSIの機能検証（RTL設計の検証）では，テスト仕様の検討，テスト環境の構築，テスト環境の動作確認，および検証の実施というプロセスが必要になります．これら一連のプロセスは，ソフトウェア開発と同じものになります．

近年，ソフトウェア開発のプロセス管理手法の一つとして，CMM（ソフトウェア能力成熟度モデル）が注目され，多くの企業がCMMを活用したプロセス管理改善活動を行っています．

ソフトウェアの品質問題は，場当たり的に対応すればするほど，ますます悪化します．品質を向上させていく道のりに近道はないと思います．それでは品質を向上させるためには何から手をつければよいのでしょうか？

その答えは，「ソフトウェア・プロセスの質」の向上にあると筆者らは考えています．ソフトウェア・プロセスとは，ソフトウェア開発に関連する作業（要求定義，設計，コーディング，テストなど）の系列のことです．質が高いソフトウェア・プロセスを実行できるプロジェクトは，結果的に品質の良いソフトウェアを効率良く作り出すことができます．

ソフトウェア・プロセスの質を向上させるためには，現在行っている開発プロセスの改善が必要になります．そのためには，まず，現在の開発プロセスにおける課題や問題点を抽出しなければなりません．しかし，この問題の抽出は，ソフトウェア開発を相当経験したベテランでなければうまく行えません．

CMMは，経験の豊富なソフトウェア技術者やマネージメントのプロたちによる，課題や問題点を抽出する観点を集めたノウハウ集と言うことができます．つまり，経験が浅く若い技術者の多いプロジェクトで，開発プロセスがあまり定まっていない場合でも，CMMの取り決めに沿ってプロセス改善を推進すれば，ある一定レベルの品質の確保が可能となるわけです．

また，品質向上のためのプロセスを良く理解しているプロジェクト・マネージャ（PM）でも，ビジネス上の理由から短期開発が多くなってきているため，各工程の期間を十分に取ることができません．この場合，納期に追われてやるべきことを簡略化したり，省いたりして開発を進めてしまうことが多くなります．これに対して，CMMはやるべきことをきちんとやるようにプレッシャをかけるしかけとして働きます．これにより品質や納期，コストについても最終的に良い結果を出すことができるのです．

以上のようにCMM導入は，品質を向上させたい場合の強い味方となります．

なお，CMMは組織レベルのプロセス改善活動なので，単独のプロジェクトでCMMを導入することは難しい面があります．しかし，CMMの考えかたは品質を向上させた

注：人材開発能力の成熟度モデルや統合プロダクト開発成熟度モデルなど，さまざまなCMMが生まれた．これらの各種CMMを統合したものは「CMMI（CMM Integration）」と呼ばれている．

い開発者のためのすばらしい教科書となることはまちがいありません.

筆者ら(富士通LSIテクノロジ)は，全社的にCMMに取り組み，各部署の各プロジェクトでCMMに従ったプロセス改善活動を実施しています．本稿では，シミュレーションを利用する大規模ディジタルLSIの機能検証にCMMを導入したケースを紹介します(下掲のコラム「LSI検証サービスとは」を参照)．導入した際の注意点や失敗した点，また期待できる効果を，実際の体験をもとに説明します.

● プロセスの質をランク付けして改善対象と課題を明示

CMMとはCapability Maturity Modelの略で，もともと米国国防総省が軍のシステム開発の委託で利用するために，米国Carnegie Mellon UniversityのSoftware Engineering Institute(SEI)に要請して開発した，ソフトウェア・プロセスの成熟度を判断する基準です.

CMMの考えかたを手短に表現すれば，個人が自己流でソフトウェアの品質を向上させることには限度があり，組織として標準的なやりかたで，だれもが同じように開発できることが必要であり，さらに，そのやりかたを継続的に改善していくことが必要だということになります．つまり，ソフトウェア開発において，ソフトウェア・プロセスの質が向上すれば，ソフトウェア開発の品質および生産性の向上につながるということです.

ソフトウェア開発プロジェクトの成功とは，品質，納期，コストの目標を達成することです．しかし，実際にすべてを満足できたプロジェクトは30％以下と言われています．大きなプロジェクトになれば成功率はもっと下がります．さらに，同じような要因でプロジェクトの失敗を繰り返すこと(失敗から学習していないこと)も問題となっています．つまり，今までの開発プロセスに問題があり，改善が必要な状態にあるのに改善しようとしていない，または改善するにしてもどのように進めたらよいのかわからない点が大きな問題なのです.

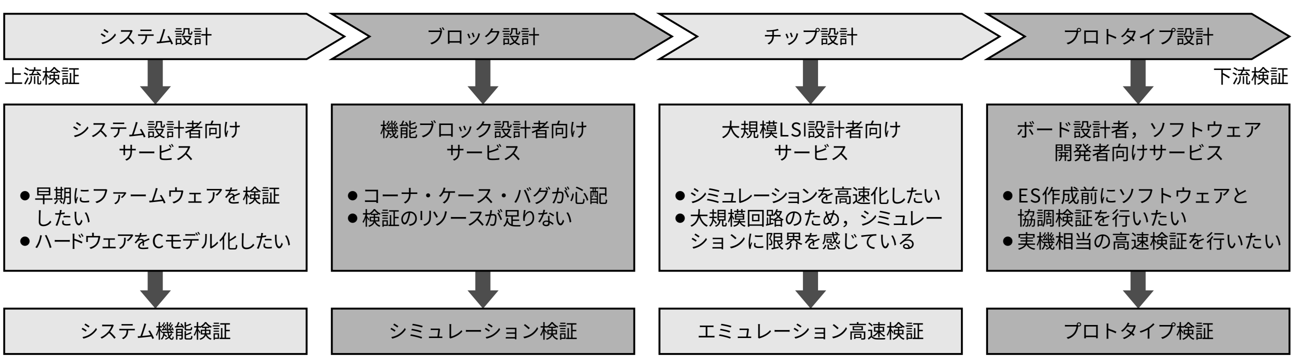
そこで登場するのがCMMです．CMMでは「ソフトウェア・プロセスの質」に成熟度レベルを設け，レベルごとに備えておかなければならない内容や要素，性質を規定しています．そのため，CMMを導入すれば，ソフトウェア・プロセスの成熟度ごとに重点的に取り組むべき改善対象の分野と課題が明確になります.

ではここで，ソフトウェア・プロセスの成熟度と，成熟度ごとに必要となる主要なプロセス群(KPA：Key Process Area；重点的に取り組むべき改善対象の分野と課題)について説明します．ソフトウェア・プロセスの成熟度には5段階の成熟度レベル(レベル1～5)が定義され，さらに各レベルで必要となるKPAが決められています．成熟度レベルと対応するKPAを**図1**に示します.

Column 1 LSI検証サービスとは

LSI検証サービスは比較的新しいビジネスなので，まだ，よくご存じない方もいるかと思います．**図A-1**に，筆者ら(富士通LSIテクノロジ)のLSI検証サービスが対象とするシステムLSIの設計工程と各工程に対する顧客の要求を示します．筆者らの場合，システムLSI設計の上流から下流までの各工程に対応して，LSI検証サービスを展開しています.

こうしたビジネスでは，LSI検証の質の向上が最重要課題となります．そのため，筆者らは全社的にCMMと取り組み，LSI検証の質の向上を目ざしています.



図A-1　LSI検証サービスの例

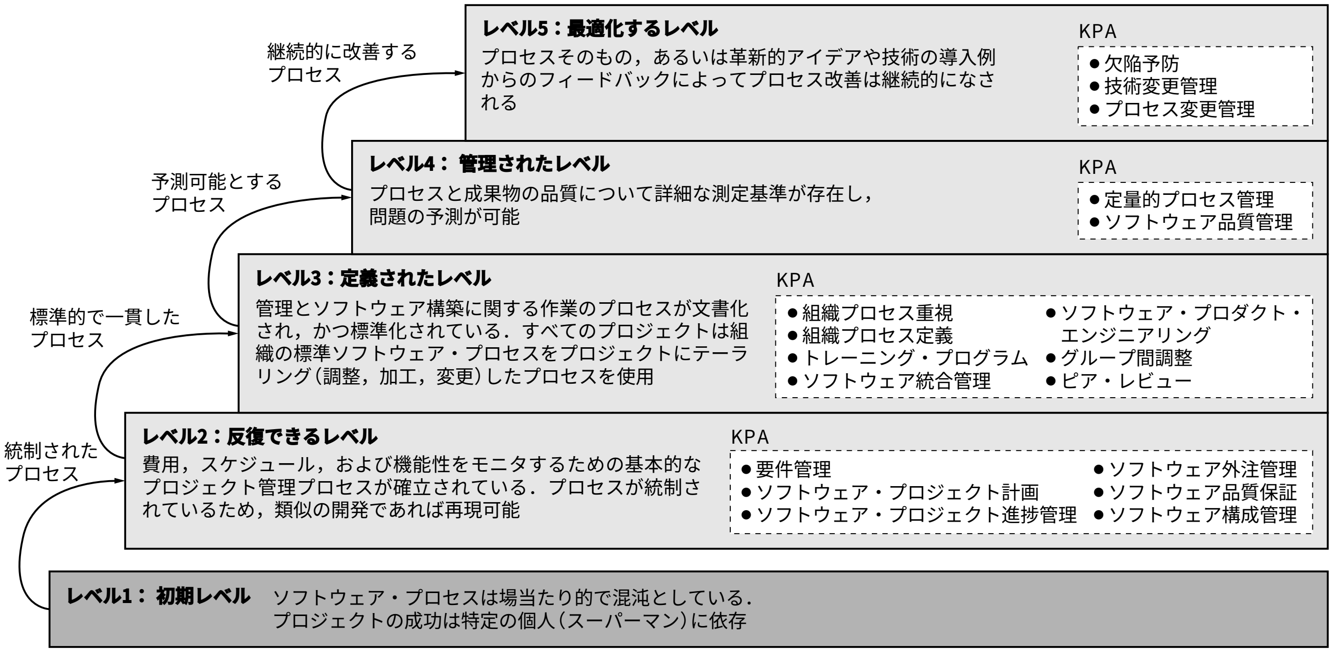


図1　CMM ソフトウェア・プロセス成熟度レベル
CMM ソフトウェア・プロセス成熟度の五つのレベルの概要と，レベルに対応するKPA（重点的に取り組むべき改善対象の分野と課題）を示す．

● 成熟度レベルは５段階

各レベルについてもう少し詳しく説明します．

1）レベル１：初期（initial）レベル

レベル1の段階では，手順らしきものはなく，場当たり的に作業が行われます．担当者のよしあしで結果が決まってしまいます．スケジュール表があっても，プロジェクト進行中に変更が多く発生したり，変更による影響範囲がほとんど見えてこないため，プロジェクトを個人の力量でなんとか遂行しているレベルです．

2）レベル２：反復できる（repeatable）レベル

レベル2は，過去の成功プロジェクトのプロセスを文字どおり「反復できる」ということになります．反復を可能とするため，スケジュールや費用，変更制御を管理し，確認できるようになっています．ただし，みずからの経験則に基づいている部分が多く，まったく新しい案件に対しては，同じように対応できるとは限りません．

3）レベル３：定義された（defined）レベル

レベル3では，組織として「開発のための標準プロセス」および「管理のための標準プロセス」を定義しています．また，組織で定義された標準プロセスを，個々のプロジェクトの内容に合わせてテーラリング（tailoring：調整，加工，変更すること）して使用することになります．レベル2ではまったく新しい案件に対応することが難しかったのですが，レベル3には標準プロセスが存在するため，新規のプロジェクトでも安定して進行させられるようになります．

4）レベル４：管理された（managed）レベル

レベル4では，収集したソフトウェア開発に関するデータを「評価」し，現在のプロセス・レベルの維持と，改善する必要のあるプロセスを見つけられるようになります．そのためには，集めるプロセス・データを正確に定義する必要があります．また，開発組織にとってどのような尺度（例えば，生産性）が有効なのかを探す段階でもあります．そのために見積もりの手段を確立することも必要になってきます．

5）レベル５：最適化する（optimizing）レベル

レベル5では，プロセスに関してすでに多くのデータを持っており，それらのデータから開発プロセスのウィーク・ポイントを見つけだし，改善する手法や技術を導入することができます．収集されたデータから，単に作業の手順を変えたり，テスト・ツールなどを導入するための根拠を得るだけでなく，必要に応じて開発組織の運用そのものの修正も行えるようになります．

● レベルアップのための改善項目をリストアップ

次に，各レベルのKPAの概要を説明します．

1）レベル２（反復できるレベル）のKPA

Column 2 エピソード1：CMM導入の効果を確認する

CMM推進委員（以下，推進）：「まずはプロジェクトが無事完了できてご苦労さま．CMMのほうも積極的に取り組んでくれて何よりです．CMMをやってみての感想を聞かせてください」

プロジェクト・マネージャ（以下，PM）：「初めてなので，最初は具体的なイメージや，何をどうするのかがなかなかつかめず，プロジェクト計画段階は内容的にも時間的にもいちばんたいへんだった．まあ，一度経験すれば，次はそれほど苦労せずにできると思う．工程完了ごとに計測事項をまとめたり，構成管理活動も定期的に実施する必要があり，不慣れな面もあってバタバタしたけど，なんとかなったかな」

検証スタッフ（以下，スタッフ）：「ピア・レビューや構成管理で作業量は増えたけど，さほど負担は感じなかった．作業の工程が並行していた時期に工程ごとに工数を計上するのは，けっこういい加減だったかも」

推進：「何でも最初はたいへんですよ，でもこの苦労が報われるときがくるでしょう，きっと．CMMを導入して良かった点は？」

PM：「これも苦労した点だけど，プロジェクト完了後，まとめとして計測項目をもとにプロジェクト完了報告書を作成することを実施したことかな．工程ごとの作業期間，工数，生産物量の予定/実績の差異分析や，ピア・レビュー結果，テスト工程の検出障害数を分類し分析することで，今後の改善点も発見できたので，そのような点は非常に良かった．次回からは完了後の作業も計画に組み入れてやっていこうかと」

スタッフ：「工程ごとにやるべきことが定まった，という点が良い点かな．今までもまったく決まってなかったわけではないけど，なんとなくグレーな部分があって，人それぞれという感じだった．それがだいたいクリアになったような．でも，ちょっとしゃくし定規な部分もあると思うけど」

● 要件管理

作業の依頼元と依頼先の間で，依頼元からの要件（依頼先での作業開始時点であらかじめ確定させておくべき事項）についての共通理解を確立し，文書化し，適切に管理します．

● ソフトウェア・プロジェクト計画

プロジェクトを管理するための妥当な計画を確立します．これらは，プロジェクト管理のためのベースになります．

● ソフトウェア・プロジェクト進捗管理

プロジェクト計画に基づいて開発の進捗を適切に管理します．問題が発生した（予測できた）場合，管理層が効果的な処置をとれるように，また，進捗状況を把握しやすいように可視性が重要になります．

● ソフトウェア外注管理

的確な外注先を選定し，それらを効果的に管理します．

● ソフトウェア品質保証

プロジェクトで用いられているプロセス，ならびに開発中の成果物について，品質を保証するためのしくみを構築し，運用・管理する手順を作り上げます．

● ソフトウェア構成管理

プロジェクトの開始から終了までの間，プロジェクトの成果物の一貫性を確立しながら維持します．また，プロジェクト関係者が成果物を適切に利用できるようなしくみを作ります．成果物（ドキュメント，プログラムなど）の版数を管理し，成果物を参照・変更する場合はつねに最新版を利用できるようにします．

2）レベル3（定義されたレベル）のKPA

● 組織プロセス重視

組織として，プロセスの開発と改善に取り組んでいく必要があります．

● 組織プロセス定義

「組織の標準プロセス」を開発し，維持・管理し，これらをつねに個々のプロジェクトでテーラリングできる状態にしておきます．

● トレーニング・プログラム

組織としてトレーニングを行う体制を体系的に整え，活動計画を立てて各要員が必要なトレーニングを受けられるようにします．

● ソフトウェア統合管理

「組織の標準ソフトウェア・プロセス」からそのプロジェクトに適合した「プロジェクトの定義されたプロセス」を作り上げます．ソフトウェア・プロジェクト計画を策定する前に必要になります．

● ソフトウェア・プロダクト・エンジニアリング

ソフトウェア成果物を効果的に作成するための開発活動を定義します．つまり，要件分析や設計，コーディング，テストなどのプロジェクトの技術的活動を定義します．

● グループ間調整

開発グループが積極的にそのほかの開発グループと連携し，協力して作業する体制や手段を確立します．

● ピア・レビュー（peer review）

早期に効率良く成果物から欠陥を取り除くために，組織的に実施する体制を整えたうえでピア・レビューを行いま

す．ピア・レビューとは，開発担当者どうしがお互いに成果物を評価するレビューです．なお，CMMにおいて単に「レビュー」という表現を使う場合，CMM活動で必要となるドキュメントや活動の状況を上級管理層やプロジェクト・マネージャが確認・承認することを指します．

3）レベル4（管理されたレベル）のKPA

● 定量的プロセス管理

プロジェクトのプロセス遂行によって達成すべき目標（工数など）を定量的に設定し，プロセスの遂行を定量的に管理します．

● ソフトウェア品質管理

プロジェクトの成果物の品質目標を定量的に設定し，実績を測定します．

4）レベル5（最適化するレベル）のKPA

● 欠陥予防

欠陥の原因を特定し，その再発を予防します．これにより，品質や生産性の向上，および納期短縮を図ることができます．

● 技術変更管理

新しい技術（ツール，手法，およびプロセス）を随時導入していけるようなプロセスの確立を目指していきます．新しい技術の導入は，品質，生産性の向上，納期短縮につながるものでなければなりません．

● プロセス変更管理

組織で使用されているプロセスを継続的に改善していくプロセスの確立を行います．言い換えれば，品質，生産性の向上，納期短縮について，より高いレベルを継続的に目指していくプロセスです．

● 従来のLSI検証プロセスには課題が山積み

では，LSI検証プロセスにCMMを導入する前と導入した後を具体的に比較してみましょう．

従来の検証プロセスを**図2**に示します．一般のソフトウェア開発とLSI検証が大きく異なる点は，最終的にソフトウェアによる何らかのシステム（あるいはシステムの一部）としての製品を成果物とするのではなく，検証対象のハードウェアの正常性を確認することが成果となる点です．

しかし，開発プロセスとしては一般のソフトウェア開発とほぼ同じで，仕様検討，実装仕様検討，実装（プログラミング），テストというプロセスになります．

それでは，品質や納期，コストを頭に入れて，従来の検証プロセスの問題点を洗い出します．問題の要因とその結果を挙げます．

● 作業開始当初の設計側と検証側の認識のずれ

検証内容や作業成果物，使用ツール，検証のゴールなどがあいまいなまま作業に着手してしまうことがあります．作業開始前に設計側と検証側で検証の進めかたについてイメージ合わせを実施していますが，合意の内容が残っていません．つまり，要件定義にあいまいさがある状態で開発を進めているのです．

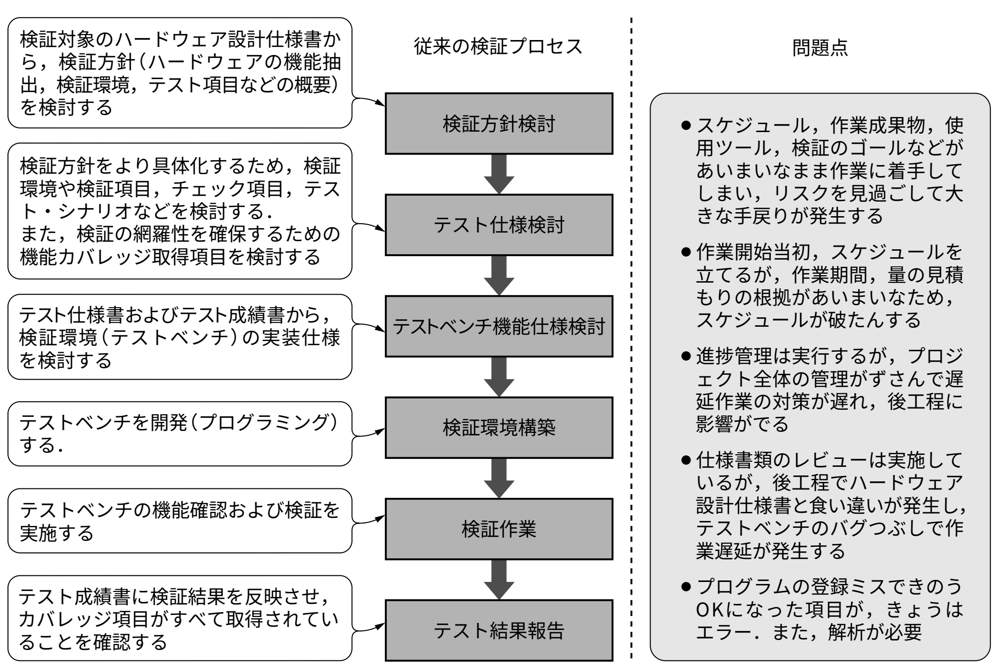


図2
従来の検証プロセスと問題点
CMM導入前の従来の検証プロセスおよび問題点を示す．

こうなると，テスト仕様の再検討，仕様書類の作り直しなどの手戻り作業が発生し，工数/スケジュールに大きく影響します．

● リスクの見過ごし

検証作業は，ハードウェア設計仕様書をバイブルとして，検証対象ハードウェア（RTL）が仕様書どおりに動作するかどうかを検証する作業です．バイブルとなるハードウェア設計仕様書の記載内容が不足しているケースが多くあります．つまり，ハードウェア設計仕様書の品質を見極めないままテスト仕様検討に着手してしまうのです．

こうなると，テスト仕様の検討段階で，検証側から設計側への問い合わせが増え，回答待ち状態が発生し，スケジュールが遅れます．

● あいまいな作業期間，作業量の見積もり

作業開始当初，スケジュールは立てますが，作業期間や作業量の見積もりの根拠があいまいなまま進めてしまうことがあります．

こうなると，スケジュールそのものが破たんします．

● 形式的な進捗管理

個々のタスクの状況は確認しますが，全体としての状況を見ていないことがあります．

こうなると，全体のゴールが見えません．

● 抜けの多いレビュー

仕様書類のレビューは実施していますが，レビューの観点が参加者によってばらばらの場合があります．

こうなると，後工程でハードウェア設計仕様書との食い違いが発生し，テストベンチのバグつぶしで作業遅延が発生します．

● 修正登録したつもりのプログラム

すでに機能確認が完了した環境で，修正したプログラムを登録し忘れたり，まちがって修正前のプログラムを登録してしまうと，プログラム全体のテストを開始したときにエラーが多発します．

こうなると，解析時に，修正したはずの箇所が修正されていないことによる手戻りが発生します．

● LSI検証プロセスにCMMを導入

CMM導入後の検証フローを**図3**に示します．

プロセスそのものに大きな変更はありませんが，各プロセスで実施するべき作業内容やその方法をより明確にすることが必要となりました．CMMを導入して変化したことを以下に述べます．

1）各プロセス内の作業内容の棚おろし

CMM（レベル3）を導入するにあたり，対応が必要となるKPAのソフトウェア・プロダクト・エンジニアリングにおいて，各プロセスの作業内容の棚おろしを行います．これによって，従来の「検証方針検討」という名称を「要件分析・定義」に変更し，作業内容を充実させました．

2）要件分析・定義

検証方針（検証対象のハードウェア設計仕様書から，検証環境および検証内容などの概要を検討したもの）を明確にしました．また，スケジュールの概略や作業成果物，開発ツール，検証ゴールを明確にしました．

さらに，設計側と要件定義書のレビューを行い，合意を取ったうえでプロジェクト計画を策定しました．なお，要件定義書は構成管理対象物とし，管理するようにしました．

加えて，不確定要素の分析を実施し，要件定義段階より「リスク管理」を実施していくようになりました．

3）プロジェクト計画

プロジェクト運営にかかわる以下の事項について計画を

Column 3　エピソード2：見せかけのCMM導入

推進：「このプロジェクトはCMM導入を積極的に行っていますね．ほぼ満点です．苦労したでしょうね」

PM：「そうですよ．作業が増えて苦労していますよ」

推進：「今の調子でがんばってください．推進側としてはうれしいかぎりです」

──CMM推進委員立ち去る

PM：「あんなこと言ってるけど，開発者の苦労も知らないで『やれやれ』って言っていればいいんだから，楽な立場だよな」

スタッフ：「そうですよね．形式的にもやっていないと課長がうるさいですからね」

PM：「来週は内部チェックがある．忙しいのに準備しなければならないな．とりあえずレビューはやったことにして，今週中にレビュー報告書を作っておいてくれよ．後で俺がチェックするから」

スタッフ：「そうですよね．まともにやってたら帰れないですもんね」

≪解説≫

開発者の意識改革が十分でなく，トップダウンでCMMの導入を進めた場合に陥るわなです．このような場合でもアセスメントは合格します．しかし，品質向上が図れないだけでなく，むだな工数を使用することによる効率ダウンおよび開発者の士気の低下という大きな問題が発生します．

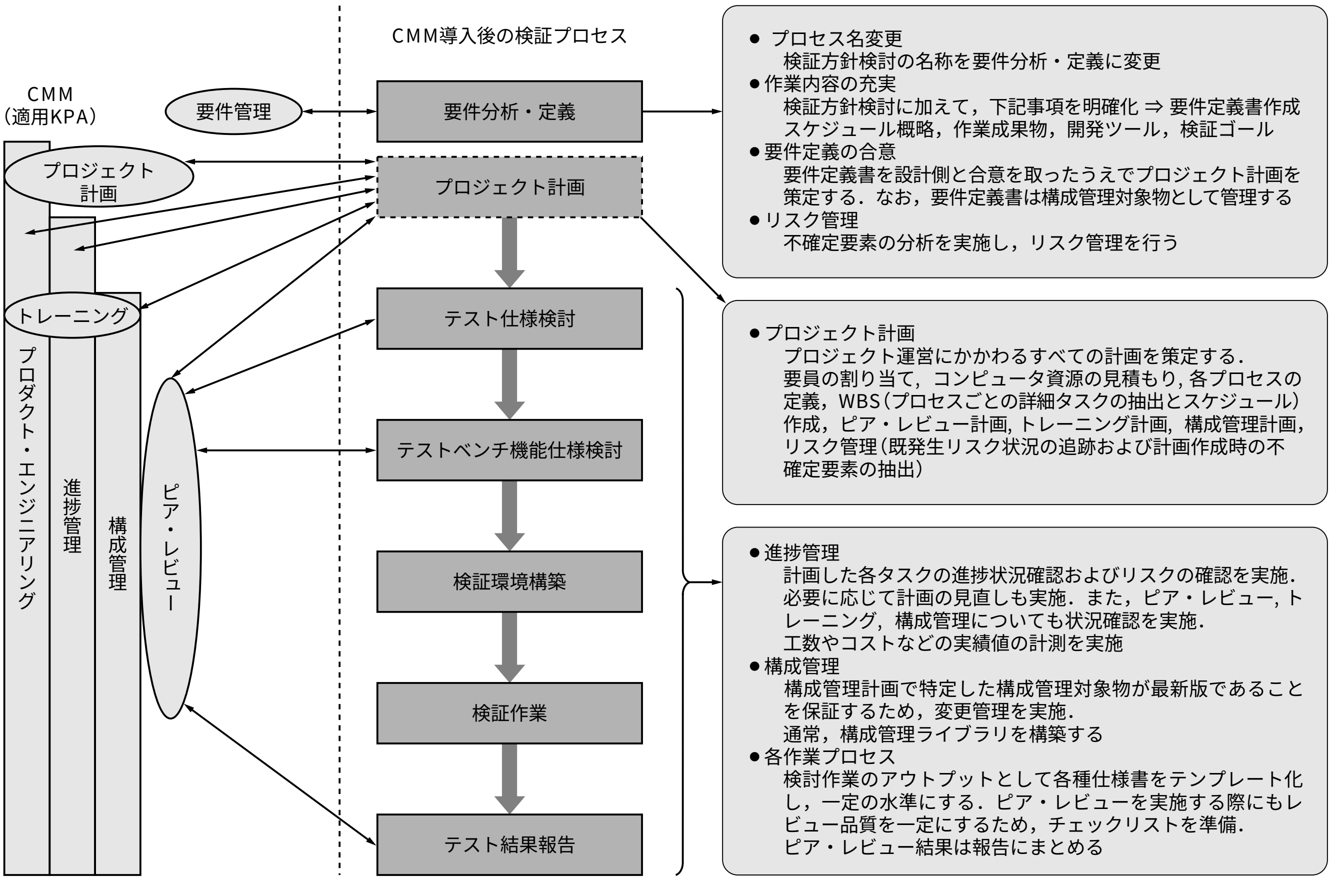


図3　CMM導入後の検証フロー
CMM導入後の検証プロセスと関連するKPAを示す．また，CMMを導入したことによるプロセス内の作業の変更点を示す．

立てるようになりました．すなわち，要員の割り当て，コンピュータ・リソースの見積もり，各プロセスの定義，WBS（work breakdown structure；プロセスごとの詳細タスクの抽出とスケジュール）作成，ピア・レビュー計画，トレーニング計画，構成管理計画，リスク管理（既発生リスク状況の追跡および計画作成時の不確定要素の抽出）です．

4）進捗管理

計画した各タスクの進捗状況の確認およびリスクの確認を実施し，必要に応じて計画を見直すようになりました．また，ピア・レビューやトレーニング，構成管理についても状況確認を実施するようになりました．さらに，工数やコストなどの実績値の計測を行うようになりました．

5）構成管理

構成管理計画で特定した構成管理対象物が最新版であることを保証するため，手順に従った変更管理を実施するようになりました．

6）各作業プロセス

検討作業のアウトプットとして，各種仕様書をテンプレート化し，水準をそろえられるようにしました．また，ピア・レビューを実施する際にも，レビュー品質を一定にするため，チェックリストを準備しました．すべての項目がチェックされたことを確認したら，レビューが完了するようにしました．ピア・レビューの結果は，指摘事項（原因分類を含む）とともに報告書にまとめるようになりました．

● CMM導入の利害得失を考える

では，CMM導入のメリットとデメリットについて説明します．

1）メリット

●開発プロセスの明確化

開発プロセスを定義し，各プロセスごとの入力や出力，手順を明確化することで，より確実に作業を進められるようになりました．

●プロジェクト計画の見積もり根拠

各開発プロセスの工数や成果物の量を一定の尺度に基づいて見積もることで，個人差による計画のばらつきを少な

Column 4 エピソード３：しゃくし定規なCMMの導入

PM：「先週頼んだ問題管理ツールの導入はできたのか？ 来週から使用するからな」
スタッフ：「まだですよ．今やっている検討資料はあすまでに提出しなければならないんです．準備する時間が取れる状態じゃないですよ．この資料が遅れてもいいんですか？」
PM：「遅れていいわけないだろう．作業を調整したまえ．時間は自分で作るものだ」
スタッフ：「ツールを導入したからといって品質に関係あるんですか？ うちのプロジェクトはあなたと私の二人ですよ．今までどおりにExcelで管理しても品質的に何の問題もないんじゃないですか？」
PM：「何を言ってるんだ．統括部のプロセスに，このツールを使用して問題管理を行うと書いてあるんだ．やるべきことはきちんとやってくれよ」

≪解説≫
きまじめなリーダが陥りやすいわなです．組織が定義したプロセスを，そのまま担当しているプロジェクトに導入しようとしています．リーダがほんとうの意味での品質向上を考えていないか，または理解できていないようです．自分のプロジェクトの特性とほんとうに品質向上が図れるのかをよく分析・理解して組織のプロセスを導入するか，またはプロジェクト独自のプロセスで進めるかをよく検討する必要があります．つまり，テーラリングすることがたいせつなのです．

くできるようになりました．

● リスク管理の励行

リスク事項の抽出ポイントが明確になりました．各開発プロセスごとにリスク管理や状況確認のサイクルを回し，プロセスが進んでも前のプロセスでリスク状況を追跡できるようになりました．そのため，取りこぼしをなくして対処できます．

● ピア・レビューの実施

ピア・レビューとして進めかたや出席者の役割分担が明確になりました．チェック・リストを使用することで，一定の品質を確保しながら効率良くレビューできるようになりました（従来も成果物などに対するレビューは行っていたが，十分とは言えなかった）．また，ピア・レビューの結果を報告書および指摘事項の管理表に記録することで，対処の確認が容易になり，連絡の漏れなどを防止できるようになりました．

● 成果物の構成管理と変更管理

構成管理対象の成果物と変更手順を明確にすることで，成果物の版数管理と変更管理を確実に実施できるようになりました．

● 品質の向上

今回定義した各プロセスを適用導入することで，プロジェクト間の品質ばらつきを少なくし，安定した品質を確保できるようになりました．今後，実績を積み重ねることで，さらなる検証環境の品質向上が図られると考えます．

2）デメリット

● プロジェクト・マネージャの負担増加

要件定義に始まり，プロジェクト開始段階の活動が多くなります．そのため，プロジェクト・マネージャの負担は確実に増えます．無論，慣れの部分はあるとは思いますが，CMM導入以前と比較して負担増となる点はやむをえないと筆者らは考えています．

作業が走り出すと，工程完了ごとに計測事項をまとめたり，構成管理活動を管理したりと，定期的なイベントにも新たに対応する必要があります．また，作業完了後も計測項目をもとにプロジェクト完了報告書を作成することが必要となり，プロジェクト完了時点でまとめ作業を行わなければなりません．

CMM導入以前と比較して負担が増えていると考えられますが，これらの作業を定型化していけば，やがてはあたりまえのしごとになると思います．見かたを変えれば，これらの負担増は本来行うべきことを行っていなかったために生じたものです．この負担と比べれば，品質向上などで得られるメリットのほうがはるかに大きいと考えています．

参考・引用*文献

(1) ソフトウェア能力成熟度モデル 1.1版（CMU/SEI-93-TR-24公式日本語版），ソフトウェア技術者協会，1999年，http://www.sea.jp/CMM/publish/CMM-J99.html
(2) 能力成熟度モデルのキープラクティス 1.1版（CMU/SEI-93-TR-25公式日本語版），ソフトウェア技術者協会，1999年，http://www.sea.jp/CMM/publish/CMM-J99.html
(3) 谷口功；図解入門よくわかる最新CMMの基本としくみ，秀和システム，2003年7月．

たかはし・まさる
富士通LSIテクノロジ（株）

<筆者プロフィール>
高橋 太．富士通LSIテクノロジ SoCソリューション統括部所属．ハードウェアのシミュレーション検証専門部隊で，プロジェクト・マネージャとCMM推進役を担当．